Hardware Versus Analytical Redundancy for Fault Detection. Application for a Servo System

Alexandar Ichtev

***Abstract*— In this paper two approaches for fault diagnosis are considered. The first investigated approach uses parity relations for fault detection and isolation. The second one is based only on the models. It relies only on an analytical redundancy. The problem is stated in its general formulation as well as for implementation – application to a servo system. From the system's model, structured residuals for fault detection and fault isolation are obtained. The main emphasis is placed on the sensor fault. However, actuator fault can be considered as well. The results are compared to the second approach based on models. The advantages and disadvantages of the two approaches are discussed. Experiments with laboratory setup – servo system are carried out. The obtained results are discussed as well.**

***Index Terms*—Fault diagnosis, Analytical redundancy, Hardware redundancy, Servo system.**

I. INTRODUCTION

In recent times, there has been increase in the demand for performance of systems working in different environments. In order to satisfy these requirements, more and more sophisticated systems with a larger number of sensors, actuators and other components are being built. As a result, the probability of a fault is increasing as well. On the other hand, there are increasing safety demands as well. In order to satisfy those requirements for automated systems, reliable fault detection methods are needed. In this paper the IFAC-Technical Committee definition of a fault is considered [1]: "A fault is an unpermitted deviation of at least one characteristic property (feature) of a system from the acceptable, usual, standard condition".

The most common faults in an automated system are sensor faults and actuator faults. They will be the focus of this paper. Especially hazardous for an automated control system is sensor fault. Even relatively small deviation from the correct measurement forces the closed loop system in undesirable operational regime. Thus, verification of the measurements is of great importance. After successful fault diagnosis and detection of a fault, the system should be govern to a safe shut down state or a fault tolerant algorithm for system reconfiguration should be applied.

In this paper special attention is paid to the reliability of the fault detection algorithms [2], [3]. The quality of the fault detection system is based on three factors. The detection time, i.e., the time after a fault appearance to its actual detection. In addition, it is of great importance to avoid false alarms, i.e., the algorithm should not detect fault when the system is fault free. Also, it is important to have no missed faults, i.e., the situation when there is a fault in the system, but the fault detection algorithm did not detect it. The cost of the system as well as additional weight and volume are in consideration since it is of great importance for some applications (especially in aero and space industries).

It is a well known fact that in time the sensors measurement can drift or can be affected from the surrounding environment. In order to verify the measurements, sensors operating according to different principles are required. The other possibility is to apply mathematical model of the plant and on its base and other measurements to reconstruct and verify the measured variable.

Alexandar Ichtev is with the Faculty of Automatics, System and Control group, FA, Technical University of Sofia, 1000 Sofia, Bulgaria (e-mail: ichtev@tu-sofia.bg).

II. REDUNDANCY

There are two fundamentally (philosophically) different types of redundancy. The first one is hardware redundancy and the second one is analytical redundancy [4]. The two principles are investigated in this paper and conclusions regarding their applicability are carried out.

According to the hardware redundancy principle, special sensors are used for fault diagnostics purposes. The most common and most simple case involves multiple sensors, which are set to measure the same variable. If only two sensors are used fault detection is achievable by monitoring the difference between the two measurements. However, fault diagnosis is not possible, because the difference does not indicate in which sensor the fault occurs. For the second part of the fault diagnosis procedure – fault isolation, additional sensor is required. Then, the decision regarding the fault situation and the correct value of the measured variable is reached after voting [4]. This approach is straightforward one and is very simple from implementtational point of view, but it requires multiple sensors, which is expensive, add additional weight and required additional space. It is also known as static redundancy of components or modules.

The other approach relies on analytical redundancy. It utilizes knowledge of the system. This can be mathematical model in form of input output relations or can be relation between two measured signals [2], [4]. In order to achieve fault diagnosis, it is necessary to perform many computations. It is essentially substitution of multiple elements with multiple computed elements.

An active fault diagnosis is essential since the next logical step is system reconfiguration which relies on data from the fault isolation to perform switching of the components or modules. The idea is to reconfigure the system in such a way that only fault free components to remain inside the

https://doi.org/10.47978/TUS.2021.71.02.015

control loop. Another aspect in the redundancy is a selection of the redundant components [2], [3] and [5]. They can be chosen to be identical or to be diverse. The advantage of the identical components is price and ease in terms of maintenance of the system. However, use of diverse components (or modules) can increase reliability of the system, since different components are not affected from the environment in the same way (one outside disturbance cannot affect all the components in the system). It is preferable to use components from different manufactures as well. In the best case scenario the redundant components operate on different physical principles.

III. Fault Detection with Parity Equations

In this paper the first chosen fault detection procedure is based on the parity equation approach which involves comparison between the measured variable with the output of a model, obtained for the nominal fault free behavior of the system [2], [5]. Then the difference between those signals is expressed as residuals. The general formulation of a residual is analytical quantity which is zero in fault-free regime and different than zero in case of a fault [4]. Thus, the residual is a quality which is indicative for a fault. In parity equations the residuals check for consistency between the plant and its model. The problem for fault detection with parity equations can be formulated with transfer function or in state space.

In this paper transfer function description of a system is used. The actuator and sensor faults are assumed to be additive faults, i.e., their effects add to the correct measurements and do not depend on the absolute value of the measured signal. An example for such fault is sensor offset. The input and output faults are noted as $f_u(s)$ and $f_y(s)$ correspondingly. The noise, acting to the plant, is expressed as equivalent output noise and it is noted with $n(s)$ (see Fig. 1). Note that they are scalars for single input single output (SISO) plants and vectors for multiple input multiple output (MIMO) plants.

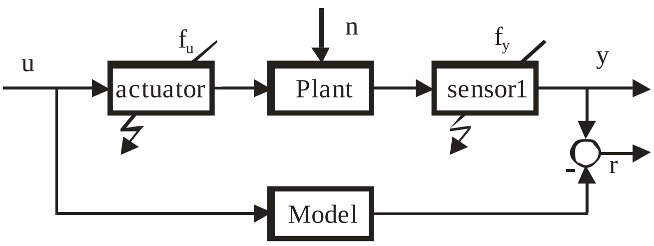

Fig. 1. Residual generation block-diagram.

If the true transfer function of the plant is noted with $G_p(s)$ and the model is presented by $G_m(s)$ (for MIMO systems they are transfer function matrixes), then the following relation can be written

$$G_p(s) = G_m(s) + \Delta G_m(s) \tag{1}$$

where $\Delta G_m(s)$ describes the modelling errors.

The used in these paper residuals are obtained from the output error equation [2], [4]. This is difference between the measured output and computed from the model

$$r(s) = y_p(s) - y_m(s) \tag{2}$$

From here it can be seen that the number of the residuals is equivalent to the number of the outputs of the system. Their number determines the number of faults that can be detected, i.e., the number of the detectable faults for this approach is equal to the number of the measurements (measured outputs of the plant).

From the definition of the transfer function, the relation between the output of the model and its input is given by

$$y_m(s) = G_m(s)u(s) \tag{3}$$

After substituting (3) into (2) the result is

$$r(s) = y_p(s) - G_m(s)u(s) \tag{4}$$

With equation (4), the residual can be obtained on the base of measurements of the inputs and outputs of a plant and by utilization of the model.

By considering that the input to the plant is formed from the input of the system and the input fault and that the output of the plant are added noise component and influence of the output fault, equation (4) can be rewritten into

$$r(s) = G_p(s)[u(s) + f_u(s)] + n(s) + f_y(s) - G_m(s)u(s) \tag{5}$$

After substituting in $G_p(s)$ from equation (1) the final form of the residual is

$$r(s) = \Delta G_m(s)u(s) + G_p(s)f_u(s) + n(s) + f_y(s) \tag{6}$$

For SISO plant there is only one residual. It can be seen that the residual (6) will be zero only if there is exact match between the model and the plant, there is no noise and there are no input or output faults. In this case separation is usually impossible. The situation drastically improves when more measurements are available, i.e., for a MIMO plants. In this case some of the elements of the residual vector deviate differently and others did not deviate at all. As this will be shown below this will help with the separation and thus fault isolation. It is known as structuring of the residual.

IV. Structured Residuals

The discussed so far residuals are directly obtained from the measurements and the plant's model. Such residuals are called primary residuals [4]. They are easy to obtain, however the information from the system can be processed and structured residuals may be calculated. These residuals can be designed in such way that special properties might be achieved. Such proportion might be decupling the residuals from some disturbances and noise. One way to take advantage is to design the residuals in such a way that the faults influenced some residuals, but they do not influence others. Then a vector or a table can be created, which shows the influence pattern. There should be at least one residual independent to a particular fault. The resulting residual patterns are also called fault signatures.

The first step in fault isolation is to assign a limit on the residual values. In the previous Section (Section 3) it was stated that the residuals are sensitive not only to the faults, but also to modelling errors and noise in the system. This means that even in fault-free working regime the residuals are different than zero. That is why the residuals are usually checked against thresholds. It is now assumed that the values inside the threshold correspond to normal operation of the system and any value outside this limit triggers the residual (alarm). This limit check yields binary outputs

$$r_i^* = \begin{cases} 0 \; if \; |r_i(t)| \leq r_{th} \\ 1 \; if \; |r_i(t)| > r_{th} \end{cases} \quad (10)$$

where r_{th} defines the threshold. $r_i^* = 1$ means that one of the thresholds is exceeded which is usually indication for a presence of a fault in the system. The thresholds are set by the designer and represent trade-off between the detection of small faults and insensitivity to modelling faults and noises. Also, special attention must be paid on the residual pattern matrix. If an error of one residual does not cause isolation of different fault, then the structured residual pattern matrix is called strongly isolating. Week isolation means that by one error another fault is isolated. An example of such pattern with two residuals is shown on Table 1.

TABLE 1
WEEK ISOLATION FAULT PATTERN

	no fault	f_1	f_2	f_3
r_1	0	1	0	1
r_2	0	0	1	1

The faults cannot be isolated if the patterns are undistinguishable. An example of such pattern with two residuals is shown on Table 2.

TABLE 2
UNDISTINGUISHABLE FAULT PATTERN

	no fault	f_1	f_2	f_3	f_4
r_1	0	1	0	0	1
r_2	0	0	1	1	1

Faults two and three have the same pattern and it is impossible to distinguish between them.

For some applications, the direction of the residuals and the size of the residual might also be considered. Some special cases are discussed in [5].

One way for residual assignment is to dedicate special residual to each particular fault. This means that only one residual is affected by a particular fault and all other are unaffected, i.e., they should have value zero. Thus, additive sensor faults can be isolated by analyzing the patterns of the residuals. An example of such pattern with three residuals is shown on Table 3 after applying the limits as shown in (10).

TABLE 3
STRONG ISOLATION FAULT PATTERN

	no fault	f_1	f_2	f_3
r_1	0	1	0	0
r_2	0	0	1	0
r_3	0	0	0	1

However, in order to generate residuals with good properties of the isolation vector, each residual should be independent at least to the faults to be isolated (one of the residuals should not be activated for that particular fault). This pattern is the best and thus this will be a goal for the first approach discussed in this paper. An example of this pattern is presented on Tab.4

TABLE 4
FAULT PATTERN FOR DIFFERENT FAULTS

	no fault	f_{y1}	f_{y2}	f_u
r_1	0	0	1	1
r_2	0	1	0	1
r_3	0	1	1	0

In order to obtain the desired structure the output error equation of a MIMO system (4) is multiplied with residual generating matrix W in order to produce the isolation vector

$$r^* = W\left[y_p(s) - G_m(s)u(s)\right] \quad (11)$$

The columns of W should be chosen in such way that they make the corresponding raw from the right-hand side independent to certain input or output. The procedure will be demonstrated on example – plant with single input and two outputs.

V. SINGLE INPUT TWO OUTPUTS SYSTEM

The investigated system is presented in Fig. 2.

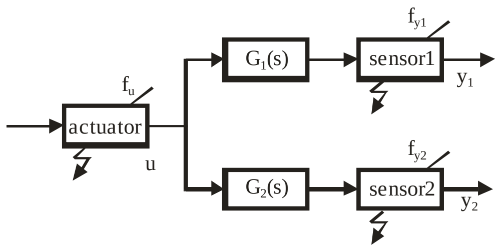

Fig. 2 Single input two output system

It is parallel system with one input and two outputs. The relations between the input and the outputs are given by

$$G_1(s) = \frac{K_1}{T_1 s+1} = \frac{y_1(s)}{u(s)} \quad (12)$$

$$G_2(s) = \frac{K_2}{T_2 s+1} = \frac{y_2(s)}{u(s)} \quad (13)$$

Equations (12) and (13) can be rewritten as

$$0 = y_1(s) - \frac{K_1}{T_1 s+1} u(s) \quad (14)$$

$$0 = y_2(s) - \frac{K_2}{T_2 s+1} u(s) \quad (15)$$

These equations can be represented in vector form

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} y_1(s) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} y_2(s) + \begin{bmatrix} -\frac{K_1}{T_1 s+1} \\ -\frac{K_2}{T_2 s+1} \end{bmatrix} u(s) \quad (16)$$

The residual generational matrix W, should have three rows in order to obtain independence of the residuals to three signal (the two outputs and the input to the system)

$$W = \begin{bmatrix} w_{y1}^T(s) \\ w_{y2}^T(s) \\ w_u^T(s) \end{bmatrix} \quad (17)$$

In order to obtain independence to $y_1(s)$, $y_2(s)$ and $u(s)$ (see equation (16)), the corresponding rows of W (computed from equation (11)) are

$$w_{y1}^T(s)\begin{bmatrix}1\\0\end{bmatrix} = 0 \Longrightarrow w_{y1}^T(s) = \begin{bmatrix}0\\1\end{bmatrix} \quad (18)$$

$$w_{y2}^T(s)\begin{bmatrix}0\\1\end{bmatrix} = 0 \Longrightarrow w_{y2}^T(s) = \begin{bmatrix}1\\0\end{bmatrix} \quad (19)$$

$$w_u^T(s)\begin{bmatrix}-\frac{K_1}{T_1s+1}\\-\frac{K_2}{T_2s+1}\end{bmatrix} = 0 \Longrightarrow w_u^T(s)\begin{bmatrix}\frac{K_2}{T_2s+1}\\-\frac{K_1}{T_1s+1}\end{bmatrix} \quad (20)$$

After multiplying the obtained matrix W with equation (16) the structured residuals are obtained

$$r_1^*(s) = y_2(s) - \frac{K_2}{T_2s+1}u(s) \quad (21)$$

$$r_2^*(s) = y_1(s) - \frac{K_1}{T_1s+1}u(s) \quad (22)$$

$$r_3^*(s) = \frac{K_2}{T_2s+1}y_1(s) - \frac{K_1}{T_1s+1}y_2(s) \quad (23)$$

These residuals have the signature presented in Table 5. From equation (21) and from Table 5 it can be seen that the first residual is independent to the first output, the second residual to the second output and the third residual to the input. From Table 5 it can be seen that the residuals are strongly isolating, because there is difference in two residuals for each fault, thus error in one residual will not going to cause wrong fault isolation.

TABLE 5
FAULT PATTERN FOR DIFFERENT FAULTS

	no fault	f_{y1}	f_{y2}	f_u
r_1^*	0	0	1	1
r_2^*	0	1	0	1
r_3^*	0	1	1	0

VI. SERVO SYSTEM

The experiments are carried out with a laboratory setup, manufactured by Inteco®. The setup is shown in Fig. 3. The control is applied to a DC motor, coupled with tachogenerator. The motor drives an inertia module, connected with backlash, magnetic break, and gearbox. The rotation of the DC motor shaft is measured with incremental encoder. The DC motor is controlled with pulse width modulation (PWM). By varying the coefficient of the PWM the effective voltage is changed according to the formula $(t) = v(t)/v_{max}$. The maximum voltage is $v_{max} = 24V$ and the control is in the range $[-1 \quad 1]$ (the sign of the PWM coefficient determines the rotational direction). The second measuring device is an incremental encoder. It is made with a disk with two set of 4096 holes in it. The first one is used to measure the rotational angle and the second one is used to determine the rotational direction. The main advantage of the system is that the two sensors are completely different. The tachogenerator is analog device based on the electromagnetic principle, while the encoder is digital sensor based on a light passing through holes. The outside disturbance such as strong magnetic field can influence one of the sensors (in this case the tachogenerator) but it will not be going to affect the other one.

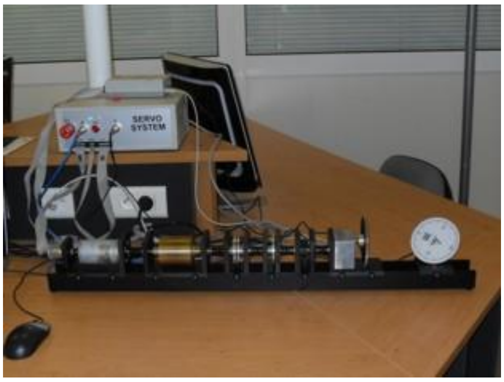

Fig. 3. Laboratory setup

VII. EXPERIMENTAL SETUP

The experiments are carried out in Matlab/Simulink® environment, with Real Time Workshop®. The block diagram of the system is presented on Fig. 4.

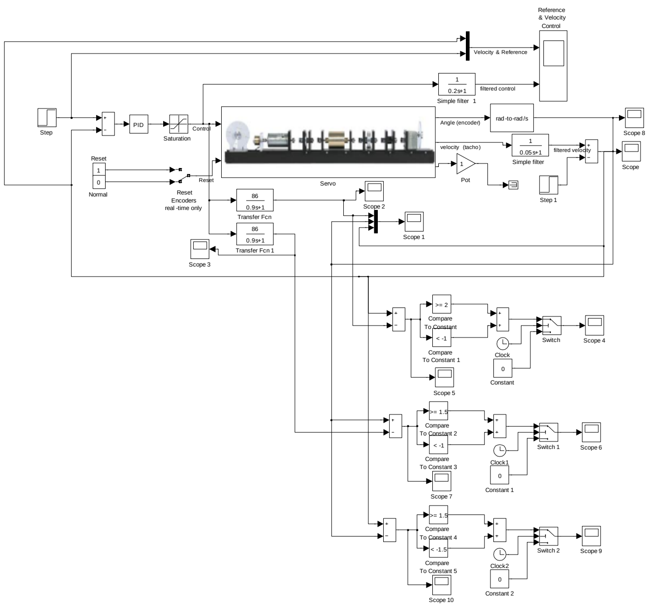

Fig. 4. Matlab/Simulink® Block-diagram for the first experiment

In the middle of the figure is shown the driver for connection to the servomechanism. It is provided by the manufacturing company Inteco®. The controller is from PI type (general PID controller is used) with coefficients $K_p = 0.06$ and $K_I = 0.03$ (the D part coefficient is 0). This controller is used for demonstrational purposes only. Similar results could be obtained with other controllers – LQR and fuzzy PI controller. They are not presented in this paper. The idea and the conclusions from these experiments are general and are applicable for all closed loop systems. On the right-hand side of Fig. 4, the angle, measured from the encoder, is transformed to angular speed, thus the reading from the tachogenerator is reconstructed.

During the first fifteen seconds of the experiment the system is in fault-free working regime. It operates in its nominal regime – following a reference - 30 rpm. At the fifteenth second an additive fault is introduced. It is sensor offset of with magnitude 5 rpm. The outputs of the system (the reading from the encoder and the output of the tachogenerator) are presented on Fig. 6.

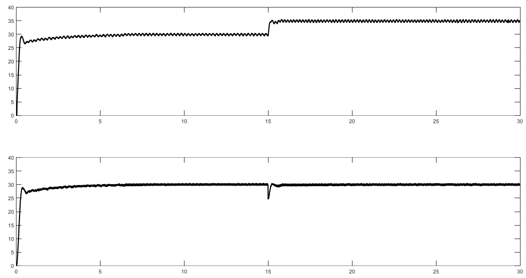

Fig. 5. Outputs from the system (on top figure the encoder and on the bottom figure from the tachogenerator)

The PID controller is compensating the faulty measurements from the tachogenerator through the feedback. The measurements from the encoder (on the top of Fig.6) are not affected by the fault and thus they are correct measurements. The negative effect of the fault can be observed in the control signal since the wrong measurements propagate through the feedback and force the system to work in undesirable operational regime.

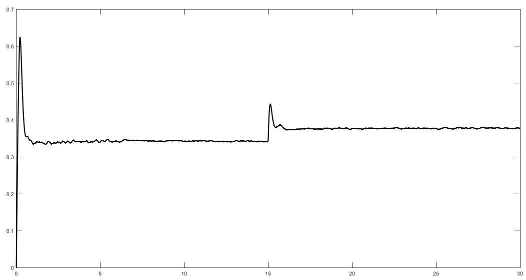

Fig. 6. Control signal of the plant

It is obvious that the controller reacts to the fault in the 15 [s]. The reaction is similar to the change in the operational conditions at the beginning of the experiment.

At the bottom of the Fig. 4, the model is represented with its transfer function and the right-hand side of equations (21)-(23) are composed. Thus, the structured residuals are modelled. On the bottom right part of the block-diagram, the thresholds are applied, and the final binary results are obtained. Also, a constant is applied in the first three seconds so the binary residuals will not be affected by the initial transient response.

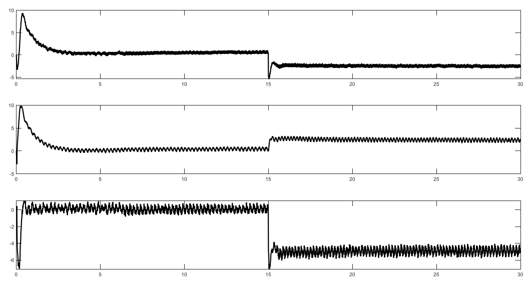

Fig. 7. Calculated residuals from (21)-(23) in Fig 4

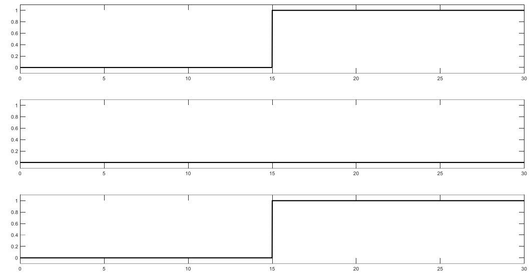

Fig. 8. Obtained binary residuals from Fig 4.

Note that the simulated fault is in the second output (the first output on Fig.4 is the encoder and the second one is the affected by the fault tachogenerator). Thus, the activation pattern is according to Tab.4.

It can be concluded that the fault diagnosis system correctly identifies the fault situation. The structured residuals have an advantage that a noise, disturbance, or other outside effect for the system will not going to influence two of the residuals and thus wrong decision regarding the fault situation is not going to be made.

VIII. MODEL BASED EXPERIMENT

The Simulink® block-diagram of the experiment is presented on Fig. 9.

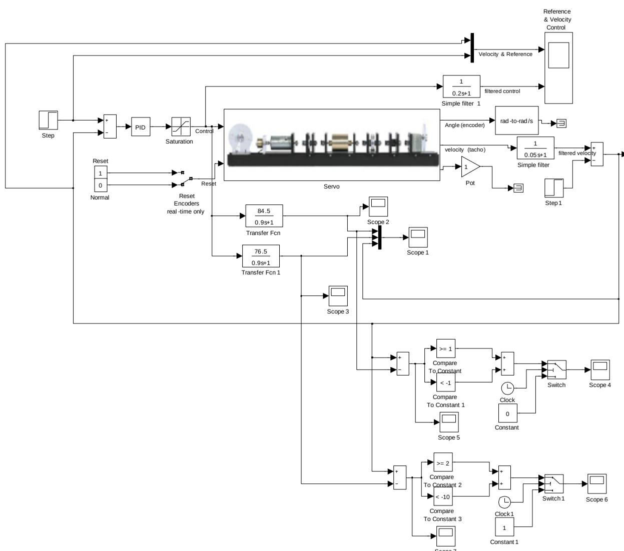

Fig. 9. Matlab/Simulink® Block-diagram for the second experiment

The second experiment is caried out with the same scenario. The difference is that only the measurement from the tachogenerator is considered. In this experiment the other sensor is not necessary. This will bring the cost of a system down, however the system will be susceptible to outside disturbances such as strong magnetic field. Note that the redundant sensor can also be extremely valuable if the systems need to continue its operation after the occurrence of a fault, i.e., in fault tolerant control scheme.

The hardware residual is substituted with a second model. In fact, in the first scheme (Fig.4) there are two models as well, however they are identical. Here the first one is modeling the systems in fault free regime and the second one is designed to model the system with the fault. It is obvious that (23) have no sense to be modeled. It will be different than zero all the time.

The output of the systems as well the outputs of the two models are presented on Fig.10. The reference signal is also placed (with dashed line). This is to be evident that the first model corresponds to fault free regime and the second one to the fault operational regime.

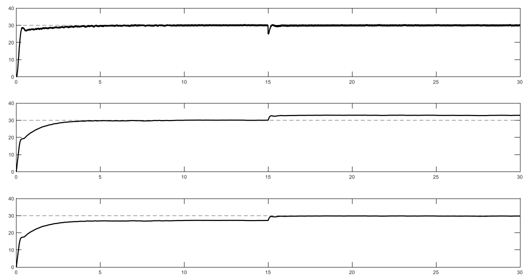

Fig. 10. Output of the system and the two models.

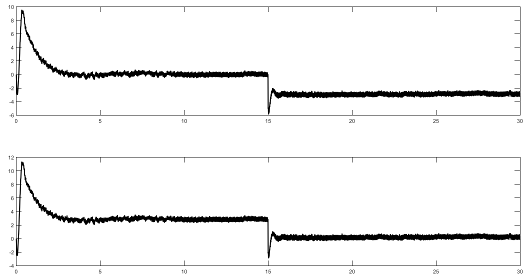

Fig. 11. Calculated residuals from the second experiment.

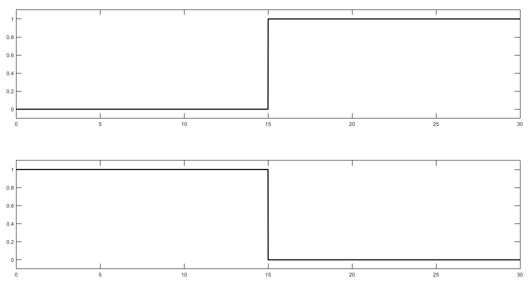

Fig. 12. Obtained binary residuals from Fig. 9.

On Fig. 11 the two residuals, i.e., the difference between the output of the system and the corresponding model is presented. It can be observed that the first residual becomes close to zero in the first 15 [s] (fault free regime) and the second one after the occurrence of the fault.

From the binary residuals presented on Fig. 12 it can be observed that the again two residuals should be changed in order to make the conclusion that the system is operating with fault (in this experiment again it is assumed that the systems operate in fault free regime for the first three seconds). This makes the fault pattern also strong isolation. It should be noted that this experiment is conducted with predetermined fault. Also, the second approach has its limitation for detecting other faults as well as applications in the fault tolerant control system since there is no redundant physical sensor in the system.

IX. Conclusion

Two methods for fault diagnosis are compared. The common element is reliability of both approaches and minimization of wrong decisions regarding the fault situation. This is achieved with strong isolational matrix. The first one is based on hardware redundancy, although the number of redundant components is brought down to a bare minimum with the utilization of a model. The second one is purely based on models (analytical redundancy). It does not require additional sensors in the systems. This approach is cost effective. However, it has limitations on the faults which can be detected and could have limited application in fault tolerant system. Both approaches are demonstrated on a laboratory setup – servo system. The results are compared and discussed.

References

[1] R. Isermann, Balle P. “Trends in the application of model-based fault detection and diagnostics in technical processes.” Control Engineering Practice, 5 (5), (1997), pp 638-652, https://doi.org/10.1016/S0967-0661(97)00053-1

[2] R. Isermann “Fault-Diagnosis Systems”, Springer 2006, https://doi.org/10.1007/3-540-30368-5

[3] Patton, R. “Fault-tolerant control: the 1997 situation.” In Prepr. IFAC SAFEPROCESS, volume 2, 1997, pp 591-596G, https://doi.org/10.1016/S1474-6670(17)42536-5

[4] Janos Gertler, Fault Detection and Diagnosis in engineering systems, Marcel Dekker, Inc., USA, 1998

[5] R. Isermann “Fault-Diagnosis Applications”, Springer 2011, https://doi.org/10.1007/978-3-642-12767-0